

DEKOMPOSISI MORFOLOGI BENTUK BINER DUA DIMENSI MENJADI POLIGON KONVEKS DENGAN PENDEKATAN HEURISTIK

Nanik Suciati, Rosdiana Rahmawati

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Email:
Email: Nanik@its-sby.edu, ossy@inf.its-sby.edu

ABSTRAK

Dalam pengolahan citra digital dan pengenalan pola, representasi bentuk merupakan masalah yang paling mendasar. Sebuah skema representasi bentuk yang bagus dapat memberi kemudahan pada analisis bentuk untuk dikembangkan pada berbagai macam pencocokan bentuk atau tugas pengenalan.

Pada Penelitian ini, dirancang perangkat lunak untuk mendekomposisi bentuk biner dua dimensi menjadi poligon konveks dengan menggunakan teori morfologi dan pendekatan secara heuristik. Proses dekomposisi dilakukan dengan menentukan aproksimasi konveks dari citra yang diolah. Penentuan aproksimasi konveks didapat dari penyusutan bentuk untuk mengambil bentuk primitif dasar, kemudian mengembangkan kembali bentuk tersebut sehingga menjadi poligon konveks. Hasil dari proses ini berupa komponen konveks dan representasinya.

Hasil uji coba membuktikan bahwa komponen konveks yang dihasilkan dari bentuk dengan tepi yang tidak banyak lekukan akan lebih sedikit daripada bentuk dengan banyak lekukan pada tepinya. Selain itu jumlah komponen konveks yang dibutuhkan untuk menghasilkan pendekatan bentuk asal, lebih sedikit bila dibandingkan dengan algoritma Wang, algoritma Pitas dan Skeleton Transform.

Kata Kunci : *bentuk biner, poligon konveks, morfologi, komponen konveks.*

1. PENDAHULUAN

Representasi atau deskripsi bentuk adalah dasar untuk proses dan analisis bentuk-bentuk yang berelasi. Telah banyak skema deskripsi bentuk yang dikembangkan selama ini. Deskripsi bentuk terstruktur adalah salah satu skema yang banyak digunakan. Dalam deskripsi bentuk terstruktur, sebuah bentuk dideskripsikan dalam suatu keadaan yang berisi bentuk-bentuk yang lebih sederhana dan hubungan antara bentuk-bentuk tersebut. Untuk membangun sebuah deskripsi bentuk terstruktur, bentuk yang diberikan didekomposisi menjadi beberapa komponen bentuk yang lebih sederhana. Komponen-komponen bentuk tersebut biasanya adalah sub-citra dari bentuk yang diberikan.

Beberapa tahun belakang ini, sejumlah algoritma dekomposisi bentuk morfologi telah diajukan. *Mathematical morphology* adalah sebuah pendekatan untuk analisis citra berorientasi bentuk. Bagaimanapun, dampak dari operasi dasar morfologi adalah dapat memberikan interpretasi sederhana menggunakan ruang geometris dari bentuk, ukuran dan lokasi. Satu keuntungan penting dari *mathematical morphology* adalah operasi dasar morfologi dapat diimplementasikan secara paralel.

Pada Penelitian ini, akan dibahas dekomposisi citra biner menjadi sekumpulan komponen poligon konveks dengan teori morfologi yang bertujuan untuk menghasilkan representasi bentuk. Ada beberapa algoritma yang telah diajukan, antara lain *morphological skeleton transform*, yaitu mendekomposisi bentuk menjadi gabungan *disk* yang

mempunyai ukuran berbeda-beda. Algoritma Pitas dan Venetsanopoulos yaitu mendekomposisi menjadi gabungan *disk* tetapi *overlap* antar disk dengan ukuran yang berbeda tidak diperbolehkan. Algoritma Wang yang menggunakan operasi rekursif morfologi juga mendekomposisi menjadi gabungan *disk* tetapi sama sekali tidak memperbolehkan *overlap* antar *disk*. Persamaan dari ketiga algoritma tersebut adalah menggunakan satu *structuring element* dalam deskripsi bentuk yang diberikan. Jika dapat menggunakan *structuring element* lebih dari satu, maka kemungkinan komponen bentuk yang dihasilkan terlihat lebih alami. Algoritma dekomposisi morfologi ini menggunakan *structuring element* lebih dari satu. Hasil dari masing-masing komponen konveks adalah merupakan subset dari citra yang diberikan dan gabungan dari konveks-konveks tersebut akan berbentuk sama seperti citra asal. Keuntungan utama dari menggunakan poligon konveks untuk mendeskripsikan komponen bentuk adalah banyak bentuk dasar berbeda yang bisa direpresentasikan sehingga hasilnya lebih akurat.

2. MATHEMATICAL MORPHOLOGY

Mathematical morphology merupakan sebuah alat untuk menggali komponen citra yang berguna untuk representasi dan deskripsi bentuk. Morfologi bisa menghasilkan batas obyek, *skeleton* (kerangka) obyek, dan *convex hull*. Sangat berguna untuk teknik pra-proses dan pasca-proses, terutama dalam *edge thinning* (menipiskan tepi) dan *pruning* (pemangkasan). Umumnya, sebagian besar operasi

morfologi berbasis pada operasi *expand* (mengembang) dan *shrink* (menyusut)[4].

Ada 2 operasi pokok morfologi, yaitu dilasi dan erosi. Transformasi ini mengandung interaksi antara citra (A) yaitu citra yang ingin diproses dan sebuah struktur set (B) yang dinamakan *structuring element*. Sebagai pengantar, akan diterangkan kembali operasi dasar teori himpunan.

Diasumsikan bahwa A dan B merupakan subset dari Z^2 . Translasi dari A oleh x , dengan notasi A_x didefinisikan :

$$A_x = \{c : c = a + x, \text{ for } a \in A\}$$

Refleksi dari B , dengan notasi \hat{B} didefinisikan :

$$\hat{B} = \{x : x = -b, \text{ for } b \in B\}$$

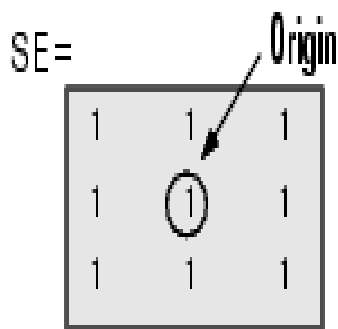
Komplemen dari A mempunyai notasi A^c , dan *difference* dari set A dan B mempunyai notasi $A - B$.

Operasi morfologi lainnya yaitu *opening* dan *closing*. Operasi ini merupakan kombinasi dari dilasi dan erosi. Berikut ini adalah persamaan operasi *opening* ($A \circ B$) dan *closing* ($A \bullet B$)

$$\begin{aligned} A \circ B &= (A \ominus B) \oplus B \\ A \bullet B &= (A \oplus B) \ominus B \end{aligned}$$

- **Structuring Element**

Merupakan faktor pengontrol pada proses dilasi dan erosi. *Structuring element* ini dapat berbentuk macam-macam, dengan daerah *origin* terletak pada pusat obyek. *Structuring element* terbagi dua macam, *flat* dan *non-flat*. Karena batasan masalah Penelitian ini berada pada citra dua dimensi, maka yang digunakan adalah *flat structuring element*. Bentuk *structuring element* ada bermacam-macam. Pada gambar 1 akan diberikan salah satu contoh bentuk *structuring element* bujur sangkar dengan ukuran 3x3.



Gambar 1 *Structuring Element*

- **Dilasi**

Dilasi adalah operasi yang mengembangkan atau menebalkan obyek. Dilasi mempunyai persamaan sebagai berikut :

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\}$$

Dimana \emptyset adalah *empty set* dan B adalah *structuring element*. Dengan kata lain, dilasi A oleh B adalah set yang terdiri dari semua lokasi *origin structuring element* dimana refleksi dan translasi dari B setidaknya saling meliputi bagian A .

- **Erosi**

Erosi merupakan kebalikan dari dilasi. Jadi operasi ini menyusutkan atau menguruskan obyek.

Erosi mempunyai persamaan sebagai berikut :

$$A \ominus B = \{z \mid (B)_z \cap A^c \neq \emptyset\}$$

Dengan kata lain, erosi A oleh B adalah suatu set dari semua lokasi *origin structuring element*, dimana translasi dari B tidak mengandung background A .

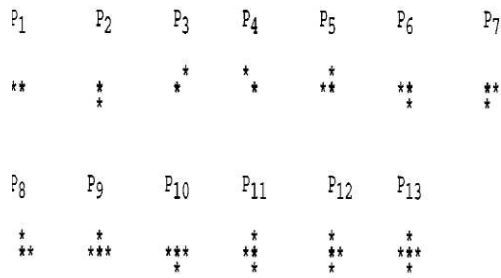
3. ALGORITMA DEKOMPOSISI MORFOLOGI

Secara garis besar, langkah pertama dalam algoritma dekomposisi morfologi ini yaitu menentukan komponen konveks C_1 . Selanjutnya untuk masing-masing komponen A_i yang saling berhubungan dalam sisa citra, yang didapat dari $A \setminus C_1$ (*difference*), tetapi terlebih dahulu diaplikasikan operasi *convex hull* untuk mengisi lubang atau bagian yang hilang yang diakibatkan dari operasi pengurangan tersebut. Lalu menentukan komponen konveks C_j untuk A_i yang telah dimodifikasi tadi. Selanjutnya semua komponen yang terhubung dari $A_i \setminus C_j$ akan teridentifikasi. Dan langkah selanjutnya akan berulang kembali sampai teridentifikasi semua komponen pembentuk citra tersebut. Komponen konveks ditentukan dengan proses aproksimasi konveks.

Dapat dikatakan bahwa sebuah poligon konveks C mempunyai dekomposisi dengan formula sebagai berikut,

$$C = (E_1 \oplus E_2 \oplus \dots \oplus E_n)_u$$

Dimana masing-masing E_n adalah salah satu dari 13 bentuk primitif dasar yang tergambar pada gambar 2. Untuk menentukan sebuah aproksimasi C untuk citra A adalah menentukan rangkaian dari bentuk primitif dasar E_1, E_2, \dots, E_n sehingga semua bentuk primitif dasar tersebut merefleksikan bentuk keseluruhan dari A .



Gambar 2 Bentuk Primitif Dasar

Algoritma ini menggunakan strategi *incremental* aproksimasi untuk membangun C . Langkah pertama yaitu menentukan E_1 untuk merefleksikan bentuk A . Kemudian E_2 ditentukan dari $A \ominus E_1$ untuk refleksi bentuk $A \ominus E_1$, dan E_3 ditentukan dari $A \ominus E_1 \ominus E_2$ untuk refleksi bentuk $A \ominus E_1 \ominus E_2$ dan seterusnya. Ada 2 dalil yang memberikan kebenaran untuk strategi yang digunakan[XU01].

Dalil 1 : misalkan E_1 adalah bentuk primitif dasar dan A_1 adalah aproksimasi konveks maksimal untuk $A \ominus E_1$. Jika E_1 I A_1 adalah komponen konveks dari A , maka tidak ada aproksimasi konveks lain untuk A yang dapat ditemukan yang mempunyai formula E_1 I X dan mengandung E_1 I A_1 sebagai subcitra yang tepat.

Dalil 2 : jika $C = A \ominus nB$ adalah poligon konveks, maka C I nB adalah aproksimasi konveks untuk A dan tidak ada aproksimasi konveks lain untuk A yang dapat ditemukan yang mempunyai formula X I nB dan mengandung C I nB sebagai subcitra yang tepat.

• Aproksimasi Konveks

Langkah pertama adalah melakukan erosi citra sampai menjadi kecil, yaitu satu langkah sebelum piksel yang ada hilang semua, dengan *structuring element* disk. Ada 2 bentuk dasar *structuring element* yang digunakan, seperti yang terlihat pada gambar 4, yaitu B_1 dan B_2 . Penggunaannya dilakukan secara bergantian. Erosi pertama dengan B_1 selanjutnya dengan B_2 lalu kembali lagi dengan B_1 dan begitu seterusnya sampai mencapai salah satu bentuk primitif dasar.

Apabila ditengah proses erosi menghasilkan bentuk poligon konveks kecil, maka proses erosi dapat langsung dihentikan. Selanjutnya tetap menentukan bentuk primitif dasarnya. Yaitu sesuai dengan rumus sebagai berikut :

$$A = A \circ D_1$$

Dimana A merupakan citra yang berbentuk konveks, dan D_1 merupakan bentuk primitif dasar. Setelah mendapat bentuk primitif dasar, maka A di-erosi dengan D_1 ,

$$A_1 = A \ominus D_1$$

Kembali lagi dari A_1 mencari bentuk primitif dasar,

$$A_1 = A_1 \circ D_2$$

Dan proses berulang terus dan didapatkan persamaan di bawah ini,

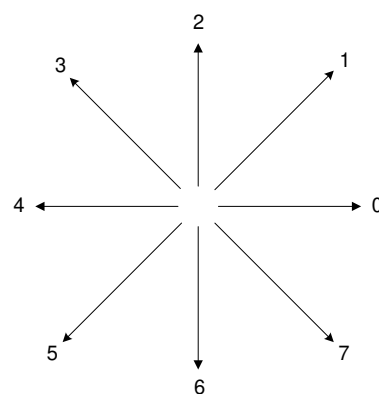
$$A_n = A_{n-1} \ominus D_n = \{u\}$$

Dimana $\{u\}$ adalah sesuatu yang tunggal (*singleton*).

Setelah mendapat bentuk primitif dasar, citra awal di-erosi dengan bentuk primitif dasar tersebut. Lalu kembali mengulang proses erosi citra sampai menjadi kecil untuk mencari bentuk primitif selanjutnya. Apabila proses erosi dengan P_n telah menghasilkan *single point*, maka *single point* tersebut di-dilasi dengan semua P_n yang telah didapat. Hasil dari dilasi tersebut menjadi bentuk aproksimasi konveks yang dicari.

• Identifikasi Bentuk Poligon

Pengidentifikasian bentuk poligon menggunakan aturan *Freeman Chain Code*[7]. Dimana suatu bentuk merupakan poligon konveks apabila normalisasi *chain code* dari bentuk tersebut merupakan urutan nilai yang meningkat secara monoton. Normalisasi dari *chain code* berupa integer dalam jumlah yang paling kecil. Contoh normalisasi dari *chain code* yang merupakan poligon konveks adalah 01234567. *Chain code* biasanya digunakan untuk merepresentasikan sebuah batas dengan cara mengambil nilai dari garis batas tersebut sesuai dengan lebar dan arah yang ditentukan, seperti pada gambar 3.



Gambar 3 8-connectivity Chain Code

• Convex Hull

Proses *convex hull* digunakan setelah operasi pengurangan (*difference*). Tujuannya adalah mengisi bagian yang hilang akibat proses pengurangan tersebut.

Pada proses *convex hull* ini dilakukan pengisian bagian yang hilang dengan cara mencari lokasi pemotongan dan menentukan ukurannya. Caranya yaitu dengan rumus sebagai berikut :

$$D' = D \cap C \oplus B_1$$

Dimana B_1 adalah *structuring element* yang terlampir pada gambar 4(a). C adalah konveks yang bersinggungan dengan citra yang terpotong. D adalah citra yang mempunyai bagian yang hilang atau terpotong karena proses *difference*. D' akan menghasilkan batas bagian yang terpotong.

Setelah mendapat batas bagian yang terpotong, batas tersebut diperluas ke semua arah, baik arah horizontal, vertikal, maupun 2 arah diagonal. Perluasan ini dilakukan dengan proses dilasi yang menggunakan *structuring element* berbentuk garis dengan ukuran 3. Pada perluasan diagonal, terkadang ada kondisi dimana terjadi kekosongan di tengah citra, untuk mengisinya maka dilakukan operasi *closing* dengan *structuring element* berbentuk bujur sangkar ukuran 2x2. Bagian yang hilang didapat dari irisan batas yang telah diperluas. Setelah mendapat hasil irisan tersebut, hasilnya digabungkan dengan citra awal yang terpotong. Dan akhirnya menghasilkan citra yang penuh.

(a)

(b)

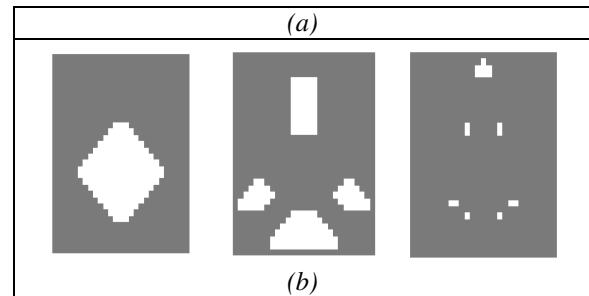
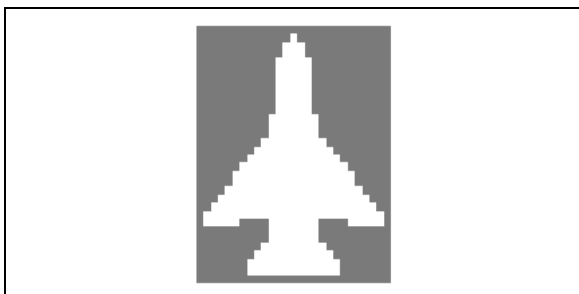
Gambar 4 (a) B_1 (b) B_2

4. HASIL UJI COBA

Uji coba ini dilakukan menggunakan komputer dengan spesifikasi sistem operasi Windows XP Home Edition, prosesor Pentium IV 1.6 GHz, memori 128 MB.

- **Uji Coba Obyek Penuh**

Uji coba dilakukan pada citra berobyek pesawat. Uji coba ini menghasilkan 12 komponen konveks.



Gambar 5 (a) Citra asal
(b) Hasil komponen konveks

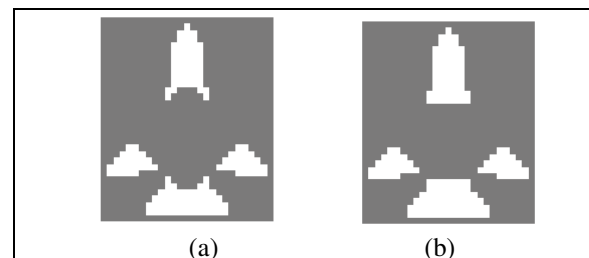
Representasi berupa string yang dihasilkan dari dekomposisi gambar 5 terlihat pada gambar 6. Pada hasil gambar 5(b) komponen konveks terpisah menjadi 3 level. Level pertama berisi c1, level kedua berisi c2 sampai dengan c5 dan level terakhir berisi c6 sampai dengan c12.

```
c1 ( c2 ( c6 ) c3 ( c8 c11 ) c4 ( c7 c9
      c10 ) c5 ( c12 ) )
```

Gambar 6 Hasil Representasi

- **Uji Coba Proses *Convex Hull***

Uji coba ini dilakukan pada citra berobyek pesawat seperti pada gambar 5.



Gambar 7 (a) Obyek terpotong akibat pengurangan (b) Hasil *convex hull*

- **Uji Coba Obyek Berlubang**

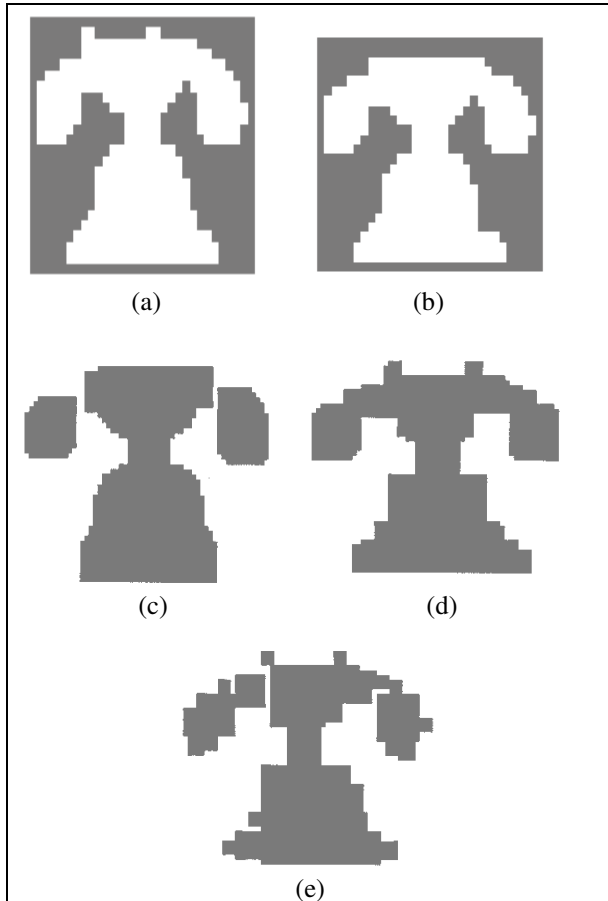
Proses yang diaplikasikan pada citra dengan obyek berlubang seperti pada gambar 8, tidak berhasil. Karena aproksimasi konveks tidak dapat teridentifikasi.



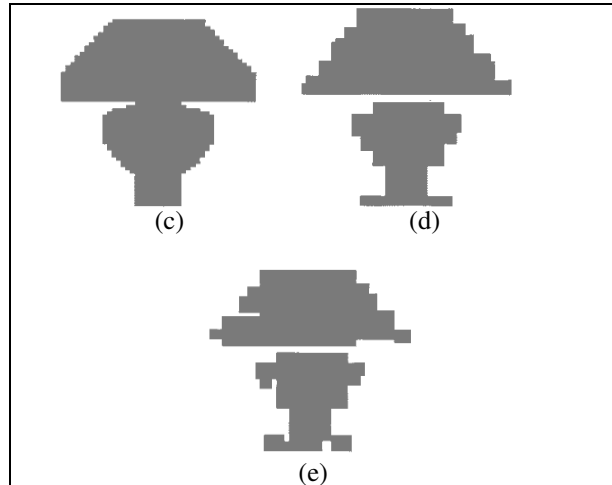
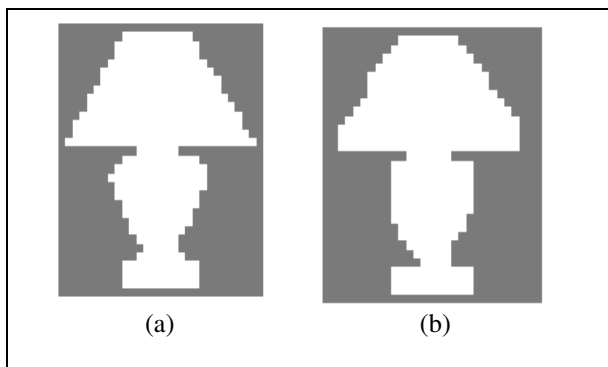
Gambar 8

Gambar Objek Berlubang

Bila hasil gabungan komponen konveks dibandingkan dengan algoritma lain, yaitu *Skeleton Transform*[3], algoritma Pitas[5], dan algoritma Wang[6], secara visual hasil dari algoritma dekomposisi morfologi ini terlihat lebih rapi. Dengan asumsi bahwa komponen konveks yang digunakan tidak mengikutsertakan komponen-komponen kecil yang telah dihasilkan.



Gambar 9 (a) Citra asal (b) Algoritma dekomposisi morfologi (c) *Skeleton transform* (d) Algoritma Pitas (e) Algoritma Wang



Gambar 10 (a) Citra asal (b) Algoritma dekomposisi morfologi (c) *Skeleton transform* (d) Algoritma Pitas (e) Algoritma Wang

Jumlah komponen konveks yang digunakan untuk membentuk citra yang dihasilkan pada gambar 9 dan gambar 10 terlihat pada tabel 1.

Tabel 1 Perbandingan jumlah komponen konveks

	Morfologi	Skeleton	Pitas	Wang
Telepon	10	62	52	30
Lampu	7	65	33	20

5. SIMPULAN DAN SARAN

Dari hasil uji coba yang telah dilakukan dapat diambil kesimpulan bahwa pada citra dengan kontur tepi yang banyak lekukan, komponen konveks yang dihasilkan akan lebih banyak, begitu juga sebaliknya. Selain itu proses dekomposisi morfologi ini tidak berhasil pada citra yang tidak penuh atau citra dengan lubang didalamnya. Secara visual, gabungan komponen konveks yang dihasilkan oleh dekomposisi morfologi ini terlihat cenderung lebih rapi bila dibandingkan dengan algoritma Wang, algoritma Pitas dan *Skeleton Transform*. Sedangkan untuk menghasilkan pendekatan bentuk citra awal, komponen konveks yang dibutuhkan oleh dekomposisi morfologi lebih sedikit daripada algoritma Wang, algoritma Pitas dan *Skeleton Transform*.

Saran pengembangan yang mungkin dilakukan untuk memaksimalkan aplikasi ini adalah yaitu membuat menu untuk pemilihan gambar mana yang ingin diproses. Sehingga masukan gambar dapat

terdiri dari beberapa obyek atau tidak hanya berobyek tunggal.

6. DAFTAR PUSTAKA

1. Gonzales, Rafael C., Woods, Richard E. *Digital Image Processing 2nd Edition*, Prentice Hall, 2001.
2. Gonzales, Rafael C., Woods, Richard E., Eddins, Steven L. *Digital Image Processing Using MATLAB*, Prentice Hall, 2004.
3. Maragos, P. A., Schafer, R. W. *Morphological Skeleton Representation and Coding of Binary Images*. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, no. 5, pp. 1228-1244, 1986.
4. Owens, Robyn, *Computer Vision IT412*, 1997; Available from : http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT3/node3.html. Accessed Mei 17, 2005.
5. Pitas, I., Venetsanopoulos, A. N. *Morphological Shape Decomposition*. IEEE Trans. Pattern Anal. Machine Intell., vol. 12, pp. 38-45, Jan. 1990.
6. Wang, D., Haese-Coat, V., Ronsin, J. *Shape Decomposition and Representation using Recursive Morphological Operation*. Pattern Recognit., vol. 28, no. 11, pp. 1783-1792, 1995.
7. Xu, Jianning. *Morphological Decomposition of 2-D Binary Shapes into Convex Polygons: A Heuristic Algorithm*. IEEE Trans. Image Processing, vol. 10, no. 1, Jan. 2001.